In [18]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
hd=pd.read_excel('Housing dataset.xlsx')
hd
```

Out[18]:

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwa
0	13300000	7420	4	2	3	yes	no	no	
1	12250000	8960	4	4	4	yes	no	no	
2	12250000	9960	3	2	2	yes	no	yes	
3	12215000	7500	4	2	2	yes	no	yes	
4	11410000	7420	4	1	2	yes	yes	yes	
...	...	...	...	...	...	...	...	...	
95	6300000	4100	3	2	3	yes	no	no	
96	6300000	9000	3	1	1	yes	no	yes	
97	6300000	6400	3	1	1	yes	yes	yes	
98	6293000	6600	3	2	3	yes	no	no	
99	6265000	6000	4	1	3	yes	yes	yes	

100 rows × 13 columns

In [19]:

```python
# 1. Find total number of furnishingstatus
hd['furnishingstatus'].count()
```

Out[19]:

```
100
```

In [20]:

```python
# 2.Find total number of furnishingstatus by prefarea
hd.groupby('prefarea').count()['furnishingstatus']
```

Out[20]:

```
prefarea
no     58
yes    42
Name: furnishingstatus, dtype: int64
```

In [21]:

```
# 3.Find total number of furnishingstatus by airconditioning
hd.groupby('airconditioning').count()['furnishingstatus']
```

Out[21]:

```
airconditioning
no     31
yes    69
Name: furnishingstatus, dtype: int64
```

In [22]:

```
# 4.Find total number of furnishingstatus by hotwaterheating
hd.groupby('hotwaterheating').count()['furnishingstatus']
```

Out[22]:

```
hotwaterheating
no     90
yes    10
Name: furnishingstatus, dtype: int64
```

In [23]:

```
# 5.Find total number of furnishingstatus bybasement
hd.groupby('basement').count()['furnishingstatus']
```

Out[23]:

```
basement
no     60
yes    40
Name: furnishingstatus, dtype: int64
```

In [24]:

```
# 6.Find total number of furnishingstatus by guestroom
hd.groupby('guestroom').count()['furnishingstatus']
```

Out[24]:

```
guestroom
no     72
yes    28
Name: furnishingstatus, dtype: int64
```

In [30]:

```
# 7.Find total of furnishingstatus by bathrooms
hd.groupby('furnishingstatus').sum()['bathrooms']
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_1860\1610625607.py:2: FutureW
arning: The default value of numeric_only in DataFrameGroupBy.sum is depr
ecated. In a future version, numeric_only will default to False. Either s
pecify numeric_only or select only columns which should be valid for the
function.
  hd.groupby('furnishingstatus').sum()['bathrooms']
```

Out[30]:

```
furnishingstatus
furnished         79
semi-furnished    62
unfurnished       35
Name: bathrooms, dtype: int64
```

In [26]:

```
# 8.Find total number of furnishingstatus by parking
hd.groupby('furnishingstatus').count()['parking']
```

Out[26]:

```
furnishingstatus
furnished         45
semi-furnished    35
unfurnished       20
Name: parking, dtype: int64
```

In [29]:

```
# 9.Find total of furnishingstatus by bedrooms
hd.groupby('furnishingstatus').sum()['bedrooms']
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_1860\1852097527.py:2: FutureW
arning: The default value of numeric_only in DataFrameGroupBy.sum is depr
ecated. In a future version, numeric_only will default to False. Either s
pecify numeric_only or select only columns which should be valid for the
function.
  hd.groupby('furnishingstatus').sum()['bedrooms']
```

Out[29]:

```
furnishingstatus
furnished         151
semi-furnished    115
unfurnished        67
Name: bedrooms, dtype: int64
```

In [28]:

```
hd.columns
```

Out[28]:

```
Index(['price', 'area', 'bedrooms', 'bathrooms', 'stories', 'mainroad',
       'guestroom', 'basement', 'hotwaterheating', 'airconditioning',
       'parking', 'prefarea', 'furnishingstatus'],
      dtype='object')
```

In [31]:

```
# 10.Find total of Price
hd['price'].sum()
```

Out[31]:

```
785924440
```

In [32]:

```
# 11.Find Average of Price
hd['price'].mean()
```

Out[32]:

```
7859244.4
```

In [33]:

```
# 12 .Find Max of Price
hd['price'].max()
```

Out[33]:

```
13300000
```

In [34]:

```
# 13.Find Mini of Price
hd['price'].min()
```

Out[34]:

```
6265000
```

In [35]:

```
# 14. Find total of area
hd['area'].sum()
```

Out[35]:

```
692757
```

In [36]:

```
# 15 . Find Average  of area
hd['area'].mean()
```

Out[36]:

```
6927.57
```

In [38]:

```
# 16. Find Max of area
hd['area'].max()
```

Out[38]:

```
16200
```

In [40]:

```
# 17.Find Mini of area
hd['area'].min()
```

Out[40]:

```
3500
```

In [42]:

```
# 18.Find total of stories by furnishingstatus
hd.groupby('furnishingstatus').sum()['stories']
```

```
C:\Users\91984\AppData\Local\Temp\ipykernel_1860\3342067483.py:2: FutureW
arning: The default value of numeric_only in DataFrameGroupBy.sum is depr
ecated. In a future version, numeric_only will default to False. Either s
pecify numeric_only or select only columns which should be valid for the
function.
  hd.groupby('furnishingstatus').sum()['stories']
```

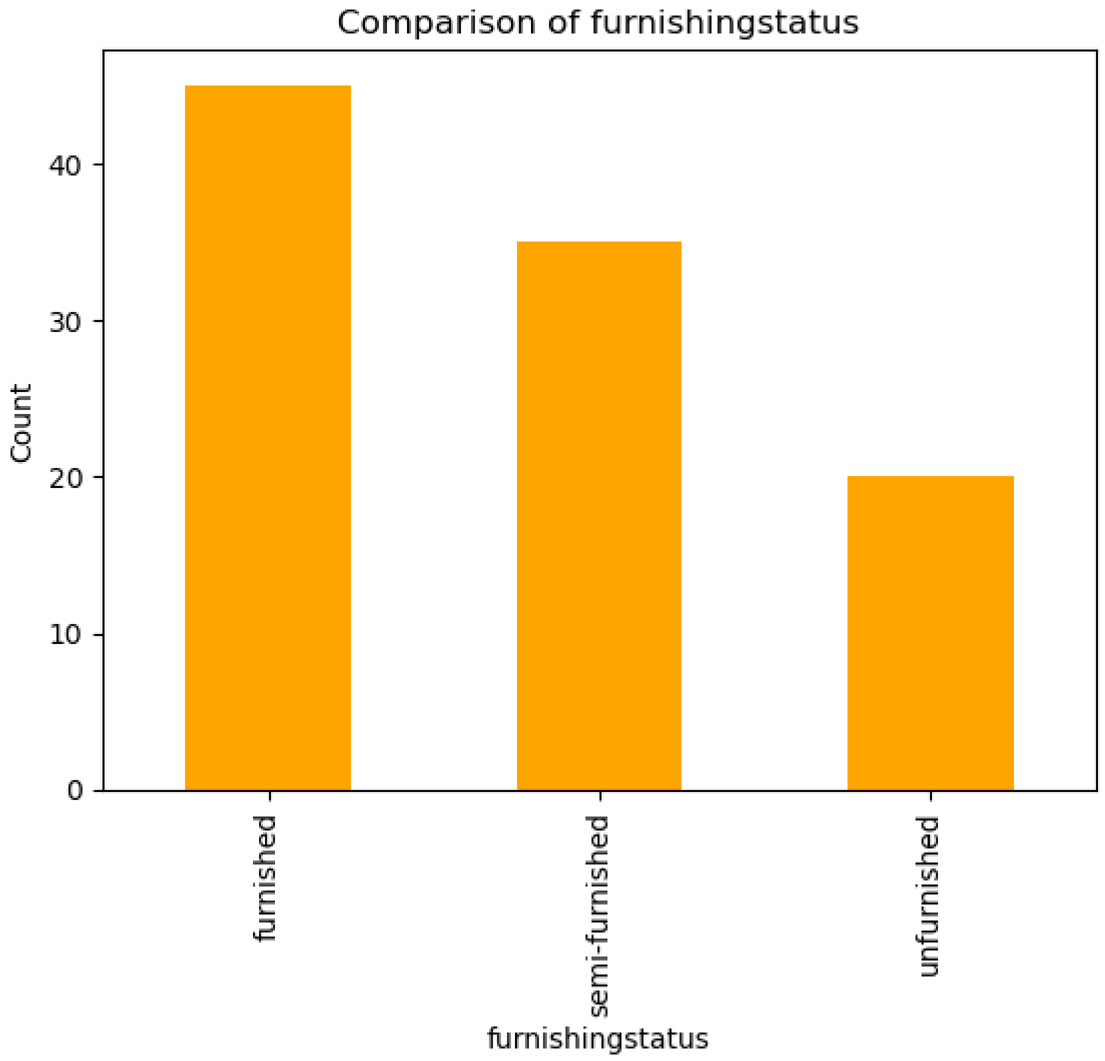
Out[42]:

```
furnishingstatus
furnished         115
semi-furnished     83
unfurnished        56
Name: stories, dtype: int64
```

In [45]:

```
# 19.create a count plot for furnishingstatus

hd['furnishingstatus'].value_counts(normalize= True)
hd['furnishingstatus'].value_counts(dropna= False).plot.bar(color='orange')
plt.title('Comparison of furnishingstatus')
plt.xlabel('furnishingstatus')
plt.ylabel('Count')
plt.show()
```

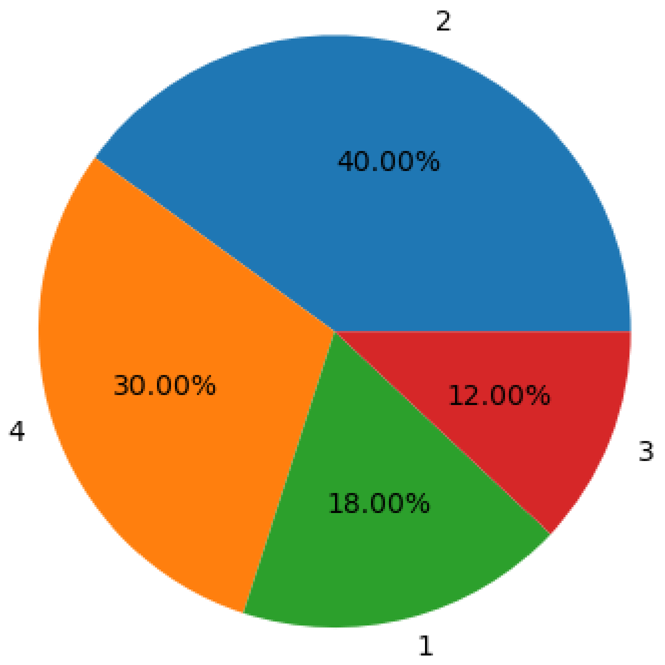


In [46]:

```
# 20.Create a pie chart to show stories
pie_label=hd.stories.value_counts().index
pie_val=hd.stories.value_counts().values
plt.pie(pie_val[:5],labels=pie_label[:5],autopct='%1.2f%%')
```

Out[46]:

```
([<matplotlib.patches.Wedge at 0x20d385ec6d0>,
  <matplotlib.patches.Wedge at 0x20d385ee620>,
  <matplotlib.patches.Wedge at 0x20d37c4fca0>,
  <matplotlib.patches.Wedge at 0x20d3860b520>],
 [Text(0.33991867422268784, 1.0461621742897658, '2'),
  Text(-1.0461621424642782, -0.3399187721714579, '4'),
  Text(0.2735590195268535, -1.0654414403595849, '1'),
  Text(1.0227541947586378, -0.40493685569926874, '3')],
 [Text(0.1854101859396479, 0.5706339132489631, '40.00%'),
  Text(-0.5706338958896062, -0.18541023936624976, '30.00%'),
  Text(0.149214010651011, -0.5811498765597735, '18.00%'),
  Text(0.5578659244138024, -0.22087464856323746, '12.00%')])
```



In [ ]: